

1.1.1 Structure and function of the processor		June1 7	June1 8	June1 9	
a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: How this relates to assembly language programs.	S1Q5d		Q1bii	**	
b) The fetch-decode-execute cycle, including its effect on registers.				***	
c) The factors affecting the performance of the CPU, clock speed, number of cores, cache.	S2Q12 *	Q1aiii	Q1a		
d) The use of pipelining in a processor to improve efficiency	S1Q5e fg S2Q3c	Q1aiii		***	
e) Von Neumann, Harvard and contemporary processor architecture.			Q1aV N		
1.1.2 Types of processor					
a) The differences between, and uses of, CISC and RISC processors.			Q2biii		
b) GPUs and their uses (including those not related to graphics)	S2Q12 *	Q1ai			
c) Multicore and parallel systems.	S2Q12 *			***	
1.1.3 Input, output and storage					
a) How different input output and storage devices can be applied as a solution of different problems.	S2Q2		Q5a		
b) The uses of magnetic, flash and optical storage devices.	S1Q2*			***	

c) RAM and ROM.	S1Q1b	Q1aii			
d) Virtual storage.				**	

1.2.1 Operating Systems					
a) The need for, function and purpose of operating systems.	S1Q10a		Q2ai		
b) Memory management (paging, segmentation and virtual memory).	S1Q1dc S2Q1a(i)	Q1bi	Q2aii+		
c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the fetch decode execute cycle.	S1Q10c S2Q1b			***	
d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time.				***	
e) Distributed, embedded, multi-tasking, multi-user and real time operating systems.		Q1b mt	7a RT		
f) BIOS.				**	
g) Device drivers.				**	
h) Virtual machines, any instance where software is used to take on the function of a machine including executing intermediate code or running an OS within another.	S1Q10b			***	
1.2.2 Applications generation					
a) The nature of applications, justifying suitable applications for a specific purpose.				*	
b) Utilities.	S1Q1a			***	
c) Open source vs Closed source.			Q5c		

d) Translators: interpreters, compilers and assemblers.	S2Q3b			***	
e) Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation).		*10d(ii)			
f) Linkers and loaders and use of libraries		10d libs			
1.2.3 Software Development					
a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application on development.	S1Q10d	Q10a, b			
b) Relative merits and drawbacks of different methodologies & when they might be used				***	
c) Writing and following algorithms.	S2Q4bc	Q7e,1 0b	Q1biii		
1.2.4 Types of Programming Languages					
a) Need for and characteristics of a variety of programming paradigms.			*Q1biv		
b) Procedural languages.				***	
c) Assembly language (including following and writing simple programs with the Little Man Computer instruction on set). See appendix 5e.	S1Q5 S2Q3	Q5a, b	Q1bi		
d) Modes of addressing memory (immediate, direct, indirect and indexed)	S1Q5b S2Q3			***	
e) Object-oriented languages (see appendix 5e for pseudocode style) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation on and polymorphism.	S1Q10e * S2Q2bc d S2Q7d		Q7bc		

1.3.1 Compression, Encryption and Hashing					
a) Lossy v lossless compression		Q3a			
b) Run Length encoding and dictionary coding for lossless compression		*Q3b			
c) Symmetric and asymmetric encryption	S2Q6c	Q4c	8bcd*		
d) Different uses of hashing	S2Q8f	Q7b			
1.3.2 Databases					
a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing. See appendix 5g	S1Q8ab	Q7a PK	Q3c PK, SK		
b) Methods for capturing, selecting, managing and exchanging data.			Q3e		
c) Normalisation to 3NF.	S1Q8c S2Q5		Q3d		
d) SQL – Interpret and modify. See appendix 5e.	S1Q8d S2Q8d- h	Q7c,d	Q3ab		
e) Referential integrity	S1Q8d S2Q5c*			***	
f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.	S2Q5c*			***	
1.3.3 Networks					
a) Characteristics of networks and the importance of protocols and standards.			4b		
b) Internet structure: <ul style="list-style-type: none"> The TCP/IP Stack. DNS . Protocol layering. LANs and WANs. Packet and circuit switching. 	S2Q8e		4a wan 4b		

			tcp/ip		
c) Network security and threats, use of firewalls, proxies and encryption		Q1cii fw			
d) Network hardware				***	
e) Client-server and Peer to peer.		Q1ci			
1.3.3 Web Technologies					
a) HTML, CSS and JavaScript.	S1Q6,1 1 S2Q8	Q9a,b, c	Q2bi		
b) Search engine indexing			Q2bii		
c) PageRank algorithm	S1Q11e			***	
d) Server and client side processing	S2Q8c			***	
1.4.1 Data Types					
a) Primitive data types, integer, real/floating point, character, string and Boolean.				***	
b) Represent positive integers in binary.		Q6ai			
c) Use of sign and magnitude and two's complement to represent negative numbers in binary.	S1Q3e S2Q11b	Q6b			
d) Addition and subtraction of binary integers.			9ab		
e) Represent positive integers in hexadecimal.	S2Q11a	Q6aii			
f) Convert positive integers between Binary Hexadecimal and denary.	S1Q3a	Q6aii	9c, d		
g) Representation and normalisation of floating point numbers in binary	S2Q11c	Q6d	9e		

h) Floating point arithmetic, positive and negative numbers, addition and subtraction	S1Q3f S2Q11d	Q6d su			
i) Bitwise manipulation and masks: shifts, combining with AND, OR and XOR	S1Q3bc d S2Q6a	Q6c	9f,g		
j) How character sets (ASCII and UNICODE) are used to represent text.	S1Q4		8a asc		
1.4.2 Data Structures					
a) Arrays (of up to 3 dimensions), records, lists, tuples.	S2Q4ab c	Q2b rec			
b) The following structures to store data: linked-list, graph (directed and undirected), <u>stack</u> , <u>queue</u> , tree, binary search tree, hash table.	S2Q7ab c	Q2a LL; Q2b HT; Q10cS T	Q5b		
C) How to create, traverse, add data to and remove data from the data structures mentioned above. (NB this can be either using arrays and procedural programming or an object-oriented			Q5b		
1.4.3 Boolean Algebra					
a) Define problems using Boolean logic.		Q4a			
b) Manipulate Boolean expressions, including use of Karnaugh maps to simplify Boolean expressions.	S2Q9a	11b	10b		
c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's laws. Distribution, association, commutation, double negation.	S2Q9b	Q4b			
d) Use logic gate diagrams and truth tables.	S1Q7ab S2Q10	Q4b,1	10a		

		1a			
e) The logic associated with D type flip flops, half and full adders	S2Q10b c	11a			
1.5.1 Computing related legislation					
a) The Data Protection Act 1998.	S1Q8d S1Q9*			***	
b) The Computer Misuse Act 1990.				***	
c) The Copyright Design and Patents Act 1988.				***	
d) The Regulation of Investigatory Powers Act 2000.	S1Q9*			***	
1.5.2 Ethical, moral and cultural issues					
a) The individual (moral), social (ethical) and cultural opportunities & risks of digital technology: <ul style="list-style-type: none"> Computers in the workforce ; Automated decision making Artificial intelligence; Environmental effects Censorship and the Internet; Monitor behavior Analyse personal information; Piracy and offensive communications Layout, colour paradigms and character sets. 		*Q8 AI	*Q6 7d 8d*		